



## ¿Qué es un Web Service?

### Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas a este nuevo video sobre servlet. Ya tenemos una estructura bastante sólida con nuestro patrón MVC. El día de mañana, cuando ustedes estén trabajando, van a utilizar algunos frameworks de más alto nivel como Spring MVC, Raptor, entre otros.

[00:20] En esos frameworks vamos a tener tal vez un controlador más sofisticado, con algunas tecnologías mejores, algunas funcionalidades que no tenemos en nuestro controlador que hemos realizado nosotros manualmente, pero no va a cambiar demasiado la funcionalidad que va a tener nuestro controlador.

[00:40] Nuestro modelo sí es lo que tal vez va a cambiar más de todo lo que hemos realizado acá, ya que nuestro modelo en este caso no tiene persistencia, o sea, hemos hecho algo solo para poder utilizar y ejemplificar, pero no estamos utilizando SQL por ejemplo para poder acceder a las bases de datos, no tenemos una base de datos, tenemos un archivo en realidad que está en memoria.

[01:05] Esas bibliotecas de más alto nivel tal vez les van a permitir poder realizar comandos SQL de una forma más fácil, usando funcionalidades de esas bibliotecas dentro de lo que es nuestro JSP, no va a haber muchas variaciones. Vamos a, básicamente dentro de lo que es el mundo Servlet con Java, lo que hemos utilizado es casi todo lo que van a ver.

[01:32] Tal vez agregar algunas otras tags, pero que sprint pueda proporcionar. Y bueno, tal vez utilicen algunos frameworks como JQuery o JavaScript para mejorar un poco más la funcionalidad. Sin embargo, está bastante sólido esa parte. Nuestras actions, una cosa buena que van a poder ver dentro de lo que son los frameworks de más alto nivel son que van a ver que, por ejemplo, en el caso de Spring MVC, van a colocar todas sus acciones en una clase solamente.

[02:04] Todas ellas van a estar dentro de una clase, incluso no van a necesitar utilizar una interfaz como nuestra acción que nosotros habíamos creado. Interfaz acción, entonces es una cosa que les va a gustar bastante. Y en el caso de los filtros, También la funcionalidad es la misma, van a tener algunos filtros ya cargados dentro de SprintMVC en este caso, que ya traen autenticación, autorización, entre otros.

[02:34] Entonces va a facilitarles mucho la vida a esos otros frameworks, pero lo más importante que quiero que comprendan es que este patrón, esta arquitectura MVC, la van a ver en todas sus aplicaciones y si no las ven en alguna de ellas, posiblemente deberían tener ese patrón, por lo menos hoy en día ese es el estándar tener ese patrón, ya que facilita mucho la manutención, el mantenimiento, perdón, de nuestras aplicaciones.

[03:05] Facilita mucho por ejemplo, cuando una persona nueva llega al equipo, facilita el poder centrarse y saber qué es lo que está viendo dentro del código. Entonces tiene muchos beneficios tener esta arquitectura. ¿Ahora qué es lo que me gustaría hablar? Me gustaría hablar que nosotros hemos estado utilizando, creando una aplicación web para servir a nuestro cliente, que en este caso sería un navegador web.

[00:00] ¿Pero qué es lo que ocurre si nuestro cliente no es un navegador web? ¿Qué pasa si nuestro cliente no entiende HTML? Por ejemplo una APP, podemos decir, bueno, una APP tiene un navegador web, podríamos desarrollar nuestra aplicación web para el navegador de nuestro celular.

[04:00] Pero 90%, perdón es un porcentaje que me inventé en realidad, pero una un gran porcentaje de veces nosotros utilizamos aplicaciones nativas o híbridas dentro de nuestros smartphones, por ejemplo, entonces esas aplicaciones nativas no entienden HTML. Lo que esas aplicaciones necesitan son los datos puros.

[04:28] Entonces esas aplicaciones no les importa que ustedes les envíen que tienen que tener ese dato en la esquina superior izquierda de la pantalla o que tiene que estar de color azul, de color verde o el tamaño. No les importa, solamente quieren saber la información datos específicos ¿Por qué?

[04:48] Porque la propia aplicación va a ser quién va a distribuir esa información de la forma que la aplicación considere mejor. Podemos enviar una lista de empresas, por ejemplo, y una aplicación podría mostrarlo verticalmente mientras que otra es las muestre horizontalmente, por ejemplo.

[05:06] Y sin embargo, nosotros le vamos a proveer los mismos datos a las dos aplicaciones. Entonces lo que yo quiero hacer es justamente hacer una lista de empresas, pero sin HTML, justamente para servir a estos clientes. ¿Y cuáles son los formatos más populares que nosotros tenemos dentro de esta idea?

[05:29] Los formatos más populares son JSON y XML. Hoy en día JSON es el más conocido. Sin embargo, XML también es conocido, ya que es utilizado en algunos proyectos más empresariales, como por ejemplo en algunos RP, son softwares bien complejos y robustos para la gestión de recursos dentro de una empresa.

[05:57] Entonces, para esos ámbitos más empresariales, XML generalmente es un poco más usado, pero en el resto de los casos ya son realmente domina bastante. Y ahora, ¿qué son estos JSON y qué son estos XML? Son strings muy muy grandes, que solamente tienen datos, nosotros no vamos a ver, por ejemplo tags de BR para poder hacer un salto ahí, un enter.

[06:24] No vamos a ver algunas tags que sean de estructuración de los datos, de cómo tienen que mostrarse. Solamente datos, llaves y valores. En el caso de por ejemplo Angular, React, VUE.js por ejemplo, hacen una especie de MVC dentro del navegador entonces es como si tuviéramos todo nuestro HTML dentro de nuestro navegador, y nosotros el servidor solamente le enviamos datos.

[06:58] Eso hace que, por ejemplo, las requests sean más rápidas, más ágiles no esté demorando tanto tiempo en tener que renderizar la página y tener que enviar toda una página completa, sino que el Angular por ejemplo pega, obtiene los datos que le enviamos y ya tiene su HTML.

[07:18] Entonces lo que hace es colocar esos datos dentro del HTML, en la posición correcta y listo, entonces es bastante más simple lo que nuestra aplicación web le envía al navegador. Entonces vamos a crear nuestro primer web service. Los web services utilizan el Protocolo http.

[07:41] Utilizando ese Protocolo http nosotros vamos a llamar funcionalidades remotamente y vamos a devolver datos en formato JSON o XML. Nosotros en el curso vamos a ver los dos formatos. Además de esto, yo les recomendaría que hagan algún curso. No necesitan hacerlo ahora, pero les recomendaría que hagan un curso que tenemos acá en Alura de HTTP.

[08:08] Perdón, antes no sé si dije Protocolo HTML, pero quise decir Protocolo HTTP. Ese curso de HTTP le va a ayudar bastante a que ustedes comprendan qué es lo que está ocurriendo por detrás. Ya que HTTP, ese protocolo está dominando el mundo, la gran mayoría de los servicios utilizan HTTP como conductor, entonces es muy recomendable que ustedes hagan ese curso ya que van a van a proveerles herramientas y van a mejorar, por ejemplo, su código, su comprensión del código.

[08:41] Y van a poder resolver problemas de una forma más fácil también gracias a que conocen ese Protocolo HTTP. Entonces ahí está mi

recomendación. Después de este curso, recomiendo hacer ese curso. Y bueno, nos vemos en el próximo video para implementar nuestro primer web Service.